View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 169 (2007) 19–32

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Behavioural Model for Klop's Calculus

Mariangiola Dezani-Ciancaglini[3, 1]

Dipartimento di Informatica, Università degli Studi di Torino
corso Svizzera 185, 10149 Torino, Italy

Makoto Tatsuta[2]

National Institute of Informatics
2-1-2 Hitotsubashi, 101-8430 Tokyo, Japan

Abstract

A model characterising strong normalisation for Klop's extension of λ-calculus is presented. The main technical tools for this result are an inductive definition of strongly normalising terms of Klop's calculus and an intersection type system for terms of Klop's calculus.

Keywords: Klop's extension of λ-calculus, strong normalisation, intersection types, inverse limit lambda models.

1 Introduction

Klop's extended λ-calculus [5] is a generalisation of Nederpelt's calculus [8]: it was introduced to infer strong normalisation from weak normalisation. We recall that *strong normalisation* means that all reductions are terminating, while *weak normalisation* means that at least one reduction to normal form is terminating. The basic idea of Klop's calculus is very simple and elegant: a redex $(\lambda x.M)N$ with x not in the free variables of M reduces to the pair $[M, N]$, instead of reducing to M. In this way no subterm is discarded, and strong normalisation coincides with weak normalisation, as proved in [5]. More precisely we use the variant of Klop's λ-calculus discussed by Boudol in [2]: we call it λ^*-calculus.

[1] e-mail: dezani@di.unito.it

[2] e-mail: tatsuta@nii.ac.jp

[3] Partially supported by FP6-2004-510996 Coordination Action TYPES and Cofin'04 project McTafi.

1571-0661 © 2007 Elsevier B.V. Open access under CC BY-NC-ND license.

doi:10.1016/j.entcs.2006.07.026

In [4] Honsell and Lenisa give the *inverse limit construction* $\mathcal{HL}_\infty$ which solves the domain equation

$$\mathcal{D} = [\mathcal{D} \to_\perp \mathcal{D}]$$

where $[\mathcal{D} \to_\perp \mathcal{D}]$ is the set of strict continuous functions from $\mathcal{D}$ to $\mathcal{D}$ (a continuous function f is strict if $f(\perp) = \perp$). In the companion paper [9] we proved that $\mathcal{HL}_\infty$ characterises strong normalisation of λ-terms. In the present paper we interpret the λ^*-calculus in $\mathcal{HL}_\infty$ and we show that $\mathcal{HL}_\infty$ also characterises strong normalisation of λ^*-terms. More precisely our results are:

- a λ^*-term S is strongly normalising iff $\mathcal{HL}_\infty$ does not interpret S as bottom in the environment which associates top to all variables;
- a λ^*-term S is persistently strongly normalising iff $\mathcal{HL}_\infty$ interprets S as top in the environment which associates top to all variables;

where a λ^*-term S is *persistently strongly normalising* if, for all n and all strongly normalising λ^*-terms $T_1, \dots, T_n$, the application $ST_1, \dots, T_n$ is strongly normalising too.

This proof is based on:

- an inductive definition of the sets of strongly normalising and persistently strongly normalising λ^*-terms;
- an extension of the intersection type assignment system of [4] for λ-terms to λ^*-terms using the rule for typing pairs of [2].

As proved in [4], we can give a finitary logical description of $\mathcal{HL}_\infty$ using intersection types. In other words, we can define the intersection type theory $\mathcal{HL}$which is the Stone dual of $\mathcal{HL}_\infty$ in the sense of [1]. This allows us to express the interpretation of a λ^*-term S in the model $\mathcal{HL}_\infty$ by means of the types derivable for S in the type system $\mathcal{HL}^*$ induced by the theory $\mathcal{HL}$.

The present paper is organised as follows. In Section 2 we introduce Klop's extended λ-calculus and we discuss the inductive definition of strongly normalising and persistently strongly normalising λ^*-terms. In Section 3 we define the model $\mathcal{HL}_\infty$ and the intersection type assignment system $\mathcal{HL}^*$, and finally we prove the characterisation results.

2 Klop's extended λ-calculus

Following [5] we extend the syntax of λ-terms with a pairing operator [,], i.e. we have the following syntax for λ^*-terms:

$$S ::= x \mid \lambda x.S \mid SS \mid [S, S].$$

Λ^* is the set of λ^*-terms.

In writing λ^*-terms we use vector notation in the standard way, i.e. $\lambda\vec{x}.S\vec{T}$ denotes $\lambda x_1 \dots x_n.ST_1 \dots T_m$, where $\vec{x}$ is $x_1, \dots, x_n$ and $\vec{T}$ is $T_1, \dots, T_m$. We use lh( ) to denote the vector length.

Following [2] we use $[S, T_1, \ldots, T_n]$ and $[S, \vec{T}]$ as short for $[\ldots[[S, T_1], T_2], \ldots T_n]$: it becomes S for $n = 0$ and $\vec{T}$ empty.

On Λ^* Boudol [2] defines the following reduction rules:

$$[\lambda x.S, U_1, \ldots, U_n]T \to_\kappa [S[x := T], U_1, \ldots, U_n]$$
$$\text{if } x \in FV(S)$$
$$[\lambda x.S, U_1, \ldots, U_n]T \to_\kappa [S, U_1, \ldots, U_n, T]$$
$$\text{if } x \notin FV(S)$$

The relation $\to_\kappa$ is the contextual closure of these rules and the relation $\to_\kappa^*$ is the reflexive, transitive closure of $\to_\kappa$.

For example $\mathbf{BO}\Delta \to_\kappa (\lambda yz.\mathbf{O}(yz))\Delta \to_\kappa \lambda z.\mathbf{O}(\Delta z) \to_\kappa \lambda z.[\lambda t.t, \Delta z] \to_\kappa \lambda z.[\lambda t.t, zz]$, where $\mathbf{B} = \lambda xyz.x(yz)$, $\mathbf{O} = \lambda vt.t$, $\Delta = \lambda u.uu$.

A λ^*-term S is a *κ-normal form* if there does not exist a λ^*-term T such that $S \to_\kappa T$.

A λ^*-term is *weakly normalising* if it has a finite reduction sequence to normal form. A λ^*-term is *strongly normalising* if all reduction sequences starting from it are finite. Let WN^* and SN^* be the set of weakly normalising and of strongly normalising λ^*-terms, respectively.

In [2] Boudol shows:

Theorem 2.1 $\mathsf{SN}^* = \mathsf{WN}^*$.

We prove that the application of nested pairs to a sequence of λ^*-terms is strongly normalising iff the application of the first element of the innermost pair to that sequence is strongly normalising and the second elements of all pairs are strongly normalising.

Lemma 2.2 $[S, \vec{T}]\vec{U} \in \mathsf{SN}^*$ *iff* $S\vec{U}, \vec{T} \in \mathsf{SN}^*$.

Proof. We will prove both directions of this claim simultaneously by induction on $\mathrm{lh}(\vec{U})$. If $\vec{U}$ is empty, the claim clearly holds. Let $\vec{U} = U\vec{W}$. By Theorem 2.1 we can assume that $S, \vec{T}, U, \vec{W}$ are κ-normal forms.

If part. If S is a λ-free term, then $[S, \vec{T}]U\vec{W}$ is a κ-normal form. Otherwise let $S = [\lambda x.V, \vec{P}]$. If $x \in FV(V)$, then there is only one possible reduction step out of $[S, \vec{T}]U\vec{W}$, i.e. $[S, \vec{T}]U\vec{W} \to_\kappa [V[x := U], \vec{P}, \vec{T}]\vec{W}$. By $SU\vec{W} \to_\kappa [V[x := U], \vec{P}]\vec{W}$, and the induction hypothesis for the if part, we conclude $[V[x := U], \vec{P}, \vec{T}]\vec{W} \in \mathsf{SN}^*$. If $x \notin FV(V)$, then there is only one possible reduction step out of $[S, \vec{T}]U\vec{W}$, i.e. $[S, \vec{T}]U\vec{W} \to_\kappa [V, \vec{P}, \vec{T}, U]\vec{W}$. By $SU\vec{W} \to_\kappa [V, \vec{P}, U]\vec{W}$ and the induction hypothesis for the only-if part, we have $[V, \vec{P}]\vec{W} \in \mathsf{SN}^*$. By the induction hypothesis for the if part, we conclude $[V, \vec{P}, \vec{T}, U]\vec{W} \in \mathsf{SN}^*$.

Only if part. If S is a λ-free term, then $SU\vec{W}$ is a κ-normal form. Otherwise let $S = [\lambda x.V, \vec{P}]$. If $x \in FV(V)$, then there is only one possible reduction step out of $SU\vec{W}$, i.e. $SU\vec{W} \to_\kappa [V[x := U], \vec{P}]\vec{W}$. By the induction hypothesis for the only-if part we conclude, since $[S, \vec{T}]U\vec{W} \to_\kappa [V[x := U], \vec{P}, \vec{T}]\vec{W}$. If $x \notin FV(V)$, then there is only one possible reduction step out of $SU\vec{W}$, i.e. $SU\vec{W} \to_\kappa [V, \vec{P}, U]\vec{W}$.

From $[S,\vec{T}]U\vec{W} \to_\kappa [V,\vec{P},\vec{T},U]\vec{W}$ we get $[V,\vec{P},\vec{T},U]\vec{W} \in \mathsf{SN}^*$. By the induction hypothesis for the only-if part, we have $[V,\vec{P}]\vec{W} \in \mathsf{SN}^*$. We conclude $[V,\vec{P},U]\vec{W} \in \mathsf{SN}^*$ by the induction hypothesis for the if part. □

We define the set PSN^* of *persistent strongly normalising* λ^*-terms as the set of λ^*-terms which preserve the strong normalisation property under application to an arbitrary number of strongly normalising λ^*-terms, i.e. $S \in \mathsf{PSN}^*$ if for all $X_1,\ldots,X_n \in \mathsf{SN}^*$ we get $SX_1\ldots X_n \in \mathsf{SN}^*$.

The pairing of λ^*-term in PSN^* with a λ^*-term in SN^* remains in PSN^* and the application of a λ^*-term in SN^* to a λ^*-term in PSN^* remains in SN^*. These are the claims of the following lemma, respectively: the proof is given in [9].

Lemma 2.3 (i) *$S \in \mathsf{PSN}^*$ and $T \in \mathsf{SN}^*$ imply $[S,T] \in \mathsf{PSN}^*$.*
(ii) *$S \in \mathsf{SN}^*$ and $T \in \mathsf{PSN}^*$ imply $ST \in \mathsf{SN}^*$.*

Similarly to [9] we also consider the class SN^*_n of λ^*-terms which preserves the strong normalisation property under application to n strongly normalising λ^*-terms, i.e. $S \in \mathsf{SN}^*_n$ if for all $X_1,\ldots,X_n \in \mathsf{SN}^*$ we get $SX_1\ldots X_n \in \mathsf{SN}^*$. Clearly $\mathsf{SN}^*_0 = \mathsf{SN}^*$.

Figure 1 defines the sets $\mathsf{PSN}^\sharp$ and $\mathsf{SN}^\sharp_n$: all rules but the last two are similar to the rules of [9] which give the inductive definition of the corresponding sets restricted to Λ. The last two rules are justified thinking that the functional behaviour of $[S,U]$ is the functional behaviour of S.

In the remaining of the present section we will show the correctness of our inductive definitions.

Theorem 2.4 $\mathsf{PSN}^\sharp = \mathsf{PSN}^*$ *and* $\mathsf{SN}^\sharp_n = \mathsf{SN}^*_n$.

To prove this, we need another theorem and a few lemmas, which we can obtain by extending the results in [9] to λ^*-terms in a straightforward way.

We call the following theorem "Substitution Theorem", since it allows to substitute different λ^*-terms in SN^*, instead of the same λ^*-term in SN^*, for different variables preserving the strong normalisation property.

Theorem 2.5 (Substitution Theorem for SN^*) *If $S[x_i := X, x_j := X] \in \mathsf{SN}^*$ for all $X \in \mathsf{SN}^*$ for all i,j $(1 \le i,j \le n)$, then $S[x_1 := X_1,\ldots,x_n := X_n] \in \mathsf{SN}^*$ for all $X_1,\ldots,X_n \in \mathsf{SN}^*$.*

Proof. The proof of the same statement for λ-terms given in [9] extends without essential changes to λ^*-terms. □

The first lemma shows a property of the set SN^*, which easily follows from Theorem 2.1 and Lemma 2.2.

Lemma 2.6 *Let $\mathrm{lh}(\vec{x}) = \mathrm{lh}(\vec{T})$, then $S[\vec{x} := \vec{T}]\vec{U} \in \mathsf{SN}^*$ and $\vec{T} \in \mathsf{SN}^*$ iff $(\lambda\vec{x}.S)\vec{T}\vec{U} \in \mathsf{SN}^*$.*

Proof. If $x \in FV(S)$, the if part clearly holds and the only-if part follows from Theorem 2.1. If $x \notin FV(S)$, from $(\lambda x.S)T \to_\kappa [S,T]$ we get the if part, and the

$$\frac{\lambda\vec{x}.S \in \mathsf{PSN}^\sharp\ (\forall S \in \vec{S}) \quad \mathrm{lh}(\vec{x}) = n \quad x \in \vec{x}}{\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^\sharp_n}$$

$$\frac{\lambda\vec{x}.S \in \mathsf{SN}^\sharp_m\ (\forall S \in \vec{S}) \quad y \notin \vec{x} \quad \mathrm{lh}(\vec{x}) = m}{\lambda\vec{x}.y\vec{S} \in \mathsf{SN}^\sharp_n}$$

$$\frac{\lambda\vec{x}.S \in \mathsf{SN}^\sharp_n \quad \mathrm{lh}(\vec{x}) = n \quad \mathrm{lh}(\vec{y}) > 0}{\lambda\vec{x}.\lambda\vec{y}.S \in \mathsf{SN}^\sharp_n}$$

$$\frac{\lambda\vec{x}.S[y := T]\vec{V} \in \mathsf{SN}^\sharp_n \quad y \in FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{SN}^\sharp_n} \qquad \frac{\lambda\vec{x}.[S,T]\vec{V} \in \mathsf{SN}^\sharp_n \quad y \notin FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{SN}^\sharp_n}$$

$$\frac{\lambda\vec{x}.S \in \mathsf{SN}^\sharp_n\ (\forall S \in \vec{S}) \quad \mathrm{lh}(\vec{x}) = n \quad y \notin \vec{x}}{\lambda\vec{x}.y\vec{S} \in \mathsf{PSN}^\sharp}$$

$$\frac{\lambda\vec{x}.S[y := T]\vec{V} \in \mathsf{PSN}^\sharp \quad y \in FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{PSN}^\sharp} \qquad \frac{\lambda\vec{x}.[S,T]\vec{V} \in \mathsf{PSN}^\sharp \quad y \notin FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{PSN}^\sharp}$$

$$\frac{\lambda\vec{x}.S\vec{T} \in \mathsf{SN}^\sharp_n \quad \lambda\vec{x}.U \in \mathsf{SN}^\sharp_m \quad \mathrm{lh}(\vec{x}) = m}{\lambda\vec{x}.[S,U]\vec{T} \in \mathsf{SN}^\sharp_n}$$

$$\frac{\lambda\vec{x}.S\vec{T} \in \mathsf{PSN}^\sharp \quad \lambda\vec{x}.U \in \mathsf{SN}^\sharp_n \quad \mathrm{lh}(\vec{x}) = n}{\lambda\vec{x}.[S,U]\vec{T} \in \mathsf{PSN}^\sharp}$$

Fig. 1. Inductive definition of $\mathsf{SN}^\sharp_n$ and $\mathsf{PSN}^\sharp$.

only-if part follows from Theorem 2.1 and Lemma 2.2. □

The following lemma, which is the key result for proving the completeness of the given inductive definition, uses in a crucial way the "Substitution Theorem" for SN^*, Theorem 2.5.

Lemma 2.7 *If $\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^*_n$, where $x \in \vec{x}$ and $lh(\vec{x}) = n$, then $\lambda\vec{x}.S \in \mathsf{PSN}^*$ for all $S \in \vec{S}$.*

Proof. For arbitrary $\vec{X} \in \mathsf{SN}^*$ with $\mathrm{lh}(\vec{X}) = n$, we have $(x\vec{S})[\vec{x} := \vec{X}] \in \mathsf{SN}^*$ by Lemma 2.6. Suppose $\mathrm{lh}(\vec{S}) = m$ and $y \notin FV(x\vec{S})$. By Theorem 2.5, $(y\vec{S})[\vec{x} := \vec{X}, y := Y] \in \mathsf{SN}^*$ holds for all $\vec{X}, Y \in \mathsf{SN}^*$. For S_i $(1 \le i \le m)$, we will show $(\lambda\vec{x}.S_i)\vec{X}\vec{Z} \in \mathsf{SN}^*$ for arbitrary $\vec{X}, \vec{Z} \in \mathsf{SN}^*$. Let Y be $\lambda\vec{z}.z_i\vec{Z}$. Then we have

$(y\vec{S})[\vec{x} := \vec{X}, y := Y] = (\lambda\vec{z}.z_i\vec{Z})\vec{S}[\vec{x} := \vec{X}] \to^*_\kappa S_i[\vec{x} := \vec{X}]\vec{Z}$. Hence $S_i[\vec{x} := \vec{X}]\vec{Z} \in \mathsf{SN}^*$. By Lemma 2.6, we have $(\lambda\vec{x}.S_i)\vec{X}\vec{Z} \in \mathsf{SN}^*$. Therefore $\lambda\vec{x}.S_i \in \mathsf{PSN}^*$. □

We can now show the soundness and completeness of the given inductive characterisations.

Proof of Theorem 2.4 [9] shows that $\mathsf{PSN}^\sharp = \mathsf{PSN}^*$ and $\mathsf{SN}^\sharp_n = \mathsf{SN}^*_n$ holds when we restrict to λ-terms, i.e. it shows $\mathsf{PSN}^\sharp \cap \Lambda = \mathsf{PSN}^* \cap \Lambda$ and $\mathsf{SN}^\sharp_n \cap \Lambda = \mathsf{SN}^*_n \cap \Lambda$. The present proof and that of [9] are similar.

We will show that *the rules generate ONLY terms which satisfy the given conditions,* that is, $\mathsf{PSN}^\sharp \subseteq \mathsf{PSN}^*$ and $\mathsf{SN}^\sharp_n \subseteq \mathsf{SN}^*_n$. This claim is proved by induction on the formation rules. It suffices to show that if the statement holds for the premises then it holds for the conclusion for each rule with $*$ instead of $\sharp$. For example for the rule

$$\frac{\lambda\vec{x}.S \in \mathsf{PSN}^\sharp\ (\forall S \in \vec{S}) \quad \mathrm{lh}(\vec{x}) = n \quad x \in \vec{x}}{\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^\sharp_n}$$

it is enough to show $(\lambda\vec{x}.x\vec{S})\vec{X} \in \mathsf{SN}^*$ for all $\vec{X} \in \mathsf{SN}^*$ of length n. By the induction hypothesis, we have $\lambda\vec{x}.S \in \mathsf{PSN}^*$. Then $S[\vec{x} := \vec{X}] \in \mathsf{PSN}^*$ by Lemma 2.6. Let $x_j = x$. By Lemma 2.3(ii), we have $X_j\vec{S}[\vec{x} := \vec{X}] \in \mathsf{SN}^*$. By Lemma 2.6, we have $(\lambda\vec{x}.x\vec{S})\vec{X} \in \mathsf{SN}^*$, so we conclude $\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^*_n$.

We also consider the rule:

$$\frac{\lambda\vec{x}.S\vec{T} \in \mathsf{SN}^\sharp_n \quad \lambda\vec{x}.U \in \mathsf{SN}^\sharp_m \quad \mathrm{lh}(\vec{x}) = m}{\lambda\vec{x}.[S,U]\vec{T} \in \mathsf{SN}^\sharp_n}$$

We assume $m \geq n$, the proof for $m < n$ being similar. By the induction hypothesis, $\lambda\vec{x}.S\vec{T} \in \mathsf{SN}^*_n$, and $\lambda\vec{x}.U \in \mathsf{SN}^*_m$. For arbitrary $\vec{X} \in \mathsf{SN}^*$ of length n, we have $(\lambda\vec{x}.S\vec{T})\vec{X} \in \mathsf{SN}^*$, and $(\lambda\vec{x}.U)\vec{X} \in \mathsf{SN}^*$. By Lemma 2.6 we get $S[\vec{x} := \vec{X}]\vec{T}[\vec{x} := \vec{X}] \in \mathsf{SN}^*$ and $U[\vec{x} := \vec{X}] \in \mathsf{SN}^*$, then by Lemma 2.2 $[S[\vec{x} := \vec{X}], U[\vec{x} := \vec{X}]]\vec{T}[\vec{x} := \vec{X}] \in \mathsf{SN}^*$. Again by Lemma 2.6 we get $(\lambda\vec{x}.[S,U]\vec{T})\vec{X} \in \mathsf{SN}^*$, so we conclude $\lambda\vec{x}.[S,U]\vec{T} \in \mathsf{SN}^*_n$.

We will show that *the rules generate ALL terms which satisfy the given conditions,* that is, $\mathsf{PSN}^\sharp \supseteq \mathsf{PSN}^*$ and $\mathsf{SN}^\sharp_n \supseteq \mathsf{SN}^*_n$. First notice that the conclusions of the given rules cover all possible shapes of λ^*-terms but $\lambda\vec{x}.x\vec{S}$ with $\mathrm{lh}(\vec{x}) = n$ and $x \in \vec{x}$ for both $\mathsf{PSN}^\sharp$ and $\mathsf{SN}^\sharp_m$ with $n < m$. This is sound since in this case we can always find λ-terms $\vec{X}$ such that $(\lambda\vec{x}.x\vec{S})\vec{X}$ does not have normal form. We refer to the proof of Theorem 4.9 in [9] for this construction.

The proof is by a double induction on the length of the longest reduction sequence to normal form and on the structure of terms. We show that if the statement holds for the conclusion then it must hold for the premises in each rule with $*$ instead of $\sharp$. The induction hypothesis applies since either the terms in the premises are obtained by reducing the term in the conclusion or they are smaller than the term in the conclusion.

The most interesting case is that of the rule

$$\frac{\lambda\vec{x}.S \in \mathsf{PSN}^\sharp\ (\forall S \in \vec{S}) \quad \mathrm{lh}(\vec{x}) = n \quad x \in \vec{x}}{\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^\sharp_n}$$

By Lemma 2.7 $\lambda\vec{x}.x\vec{S} \in \mathsf{SN}^*_n$ implies $\lambda\vec{x}.S \in \mathsf{PSN}^*$ for all $S \in \vec{S}$.

Another interesting case is the rule:

$$\frac{\lambda\vec{x}.S\vec{T} \in \mathsf{SN}^\sharp_n \quad \lambda\vec{x}.U \in \mathsf{SN}^\sharp_m \quad \mathrm{lh}(\vec{x}) = m}{\lambda\vec{x}.[S, U]\vec{T} \in \mathsf{SN}^\sharp_n}$$

whose proof follows from Lemma 2.2. The proof for the last rule is similar.

□

3 The model $\mathcal{HL}_\infty$

We start by recalling the definition of the $\mathcal{D}_\infty$-model $\mathcal{HL}_\infty$ introduced in [4] to analyse perpetual strategies in λ-calculus.

Let $\mathcal{D}_0$ be the three point lattice $\bot \sqsubseteq \mathtt{s} \sqsubseteq \top$ and $\mathcal{D}_1 = [\mathcal{D}_0 \to_\bot \mathcal{D}_0]$ be the set of strict continuous functions from $\mathcal{D}_0$ to $\mathcal{D}_0$, where a continuous function f is strict if $f(\bot) = \bot$. Moreover let $\mathtt{i}_0$ be the initial projection defined by:

$$\mathtt{i}_0(\bot) = \bot \Rightarrow \bot \qquad \mathtt{i}_0(\mathtt{s}) = \top \Rightarrow \mathtt{s} \qquad \mathtt{i}_0(\top) = \mathtt{s} \Rightarrow \top,$$

where $\mathrm{d}_1 \Rightarrow \mathrm{d}_2$ denotes the step function defined by

$$(\mathrm{d}_1 \Rightarrow \mathrm{d}_2)(\mathrm{e}) = \text{ if } \mathrm{e} \sqsupseteq \mathrm{d}_1 \text{ then } \mathrm{d}_2 \text{ else } \bot.$$

The *inverse limit construction* $\mathcal{HL}_\infty$ obtained starting from $\mathcal{D}_0$ and $\mathtt{i}_0$ is a model of the $\lambda\mathbf{I}$-calculus and of the $\lambda\mathsf{N}\mathbf{K}$-calculus as shown in [4]. The interpretation of λ-terms in $\mathcal{HL}_\infty$ is defined in the standard way:

$$\begin{aligned}
[\![x]\!]_\rho &= \rho(x)\\
[\![MN]\!]_\rho &= \mathtt{F}[\![M]\!]_\rho[\![N]\!]_\rho\\
[\![\lambda x.M]\!]_\rho &= \mathtt{G}(\lambda\!\!\lambda\, \mathrm{d} \in \mathcal{HL}_\infty.\ \text{if } \mathrm{d} \neq \bot \text{ then } [\![M]\!]_{\rho[\,\mathrm{d}/x]} \text{ else } \bot),
\end{aligned}$$

where $(\mathtt{F}, \mathtt{G})$ is a strict retraction from $[\mathcal{HL}_\infty \to_\bot \mathcal{HL}_\infty]$ to $\mathcal{HL}_\infty$. We recall that a pair of functions (f, g) is a strict retraction from $\mathcal{D}$ to $\mathcal{E}$ if they satisfy all the following conditions: f and g are continuous; $f : \mathcal{E} \to \mathcal{D}$; $g : \mathcal{D} \to \mathcal{E}$; $f \circ g = \mathsf{id}_\mathcal{D}$; $g \circ f(\bot_\mathcal{E}) = \bot_\mathcal{E}$.

We can easily extend the interpretation to λ^*-terms by the clause:

$$[\![[S, T]]\!]_\rho = \text{ if } [\![T]\!]_\rho \neq \bot \text{ then } [\![S]\!]_\rho \text{ else } \bot.$$

This clause is quite natural in view of the fact that S is the meaningful term in $[S, T]$, while T is only recorded since it could have an infinite computation.

As proved in [4], we can give a finitary logical description of $\mathcal{HL}_\infty$ using intersection types. In other words we can define an intersection type theory $\mathcal{HL}$ which is the Stone dual of $\mathcal{HL}_\infty$ in the sense of [1].

$$\sigma \leq \sigma \cap \sigma \qquad \sigma \cap \tau \leq \sigma \qquad \sigma \cap \tau \leq \tau$$

$$\sigma \leq \sigma',\ \tau \leq \tau' \Rightarrow \sigma \cap \sigma' \leq \tau \cap \tau'$$

$$\sigma' \leq \sigma,\ \tau \leq \tau' \Rightarrow \sigma \to \tau \leq \sigma' \to \tau'$$

$$(\sigma \to \tau) \cap (\sigma \to \zeta) \leq \sigma \to \tau \cap \zeta$$

$$\varphi \sim \omega \to \varphi \qquad \omega \sim \varphi \to \omega \qquad \omega \leq \varphi$$

$$\sigma \leq \sigma \qquad \sigma \leq \tau,\ \tau \leq \zeta\ \Rightarrow \sigma \leq \zeta$$

Fig. 2. Type preorder

The *set of types* of $\mathcal{HL}$ is build out of the constants φ and ω by the arrow and intersection constructors:

$$\tau ::= \varphi \mid \omega \mid \tau \to \tau \mid \tau \cap \tau.$$

We define a preorder relation on types whose axioms and rules are justified by:

- viewing "$\to$" as the function space constructor and "$\cap$" as the set intersection,
- considering the types φ and ω in correspondence with the elements $\mathtt{s}, \top$, respectively, but reversing the partial order in $\mathcal{HL}_\infty$ (this correspondence will be made explicit by the mapping $\mathtt{m}$ defined below).

Figure 2 defines the preorder $\leq$: we write $\tau \sim \sigma$ as short for $\tau \leq \sigma$ and $\sigma \leq \tau$. Notice that ω and φ are the smallest and the biggest types, respectively.

We recall that *filters* of types are sets of types upper closed and closed under intersection. Let $\mathcal{F}$ be the set of all filters: it is easy to check that $\mathcal{F}$ is an ω-algebraic complete lattice with respect to set theoretic inclusion, whose bottom element is the empty set and whose top element is the set of all types. Moreover as shown in [4] $\mathcal{F}$ is isomorphic to $\mathcal{HL}_\infty$ through the mapping:

$$\hat{\mathtt{m}}(X) = \bigsqcup_{\tau \in X} \mathtt{m}(\tau)$$

where $\mathtt{m}(\varphi) = \mathtt{s}$, $\mathtt{m}(\omega) = \top$, $\mathtt{m}(\tau_1 \to \tau_2) = \mathtt{m}(\tau_1) \Rightarrow \mathtt{m}(\tau_2)$, $\mathtt{m}(\tau_1 \cap \tau_2) = \mathtt{m}(\tau_1) \sqcup \mathtt{m}(\tau_2)$.

We extend the intersection type assignment system of [4] to λ^*-terms: we call $\mathcal{HL}^*$ the resulting system. We use Γ to denote a basis, i.e. a mapping from variables to types. The typing rules are shown in Figure 3: they are standard, but the typing rule for pairs of λ^*-terms which is given in [2]. We denote by $\vdash$ derivability in this system. It is easy to verify that strengthening and weakening are admissible rules in this system:

$$\frac{\Gamma, x{:}\sigma \vdash S{:}\tau \quad x \notin FV(S)}{\Gamma \vdash S{:}\tau} \qquad\qquad \frac{\Gamma \vdash S : \tau \quad x \notin \Gamma}{\Gamma, x{:}\sigma \vdash S : \tau}$$

The type assignment system enjoys a Generation Lemma whose restriction to Λ is proved in [3]. The proof of the last clause which is the only new clause follows easily by induction on deductions.

$$(\mathrm{Ax})\ \frac{(x\!:\!\sigma) \in \Gamma}{\Gamma \vdash x\!:\!\sigma} \qquad ([\,,\,])\ \frac{\Gamma \vdash S : \tau \quad \Gamma \vdash T : \sigma}{\Gamma \vdash [S,T] : \tau}$$

$$(\to \mathrm{I})\ \frac{\Gamma, x\!:\!\sigma \vdash S : \tau}{\Gamma \vdash \lambda x.S : \sigma \to \tau} \qquad (\to \mathrm{E})\ \frac{\Gamma \vdash S : \sigma \to \tau \quad \Gamma \vdash T : \sigma}{\Gamma \vdash ST : \tau}$$

$$(\leq)\ \frac{\Gamma \vdash S : \sigma \quad \sigma \leq \tau}{\Gamma \vdash S : \tau} \qquad (\cap \mathrm{I})\ \frac{\Gamma \vdash S : \sigma \quad \Gamma \vdash S : \tau}{\Gamma \vdash S : \sigma \cap \tau}$$

Fig. 3. Typing rules

Lemma 3.1 (Generation Lemma) (i) $\Gamma \vdash x\!:\!\tau$ *iff there is* σ *such that* $x\!:\!\sigma \in \Gamma$ *and* $\sigma \leq \tau$.

(ii) $\Gamma \vdash ST\!:\!\tau$ *iff there is* σ *such that* $\Gamma \vdash S\!:\!\sigma \to \tau$ *and* $\Gamma \vdash T\!:\!\sigma$.

(iii) $\Gamma \vdash \lambda x.S\!:\!\sigma \to \tau$ *iff* $\Gamma, x\!:\!\sigma \vdash S\!:\!\tau$.

(iv) $\Gamma \vdash [S,T]\!:\!\tau$ *iff* $\Gamma \vdash S\!:\!\tau$ *and there is* σ *such that* $\Gamma \vdash T\!:\!\sigma$.

We can now formulate Stone duality for the model $\mathcal{HL}_\infty$ generalising the result proved in [4] for λ-terms.

Theorem 3.2 (Stone Duality) *Let* $\Gamma \models \rho$ *if* $x : \sigma \in \Gamma$ *implies* $\mathtt{m}(\sigma) \sqsubseteq \rho(x)$. *We have*

$$[\![S]\!]_\rho = \bigsqcup\{\mathtt{m}(\tau) \mid \exists\Gamma \models \rho.\Gamma \vdash S\!:\!\tau\}.$$

Proof. The proof is by induction on S. The same statement restricted to λ-terms is proved in [4]. Therefore we only need to consider the case of pairs, i.e. let $S = [T,V]$. By the induction hypothesis $[\![T]\!]_\rho = \bigsqcup\{\mathtt{m}(\tau) \mid \exists\Gamma \models \rho.\Gamma \vdash T : \tau\}$ and $[\![V]\!]_\rho = \bigsqcup\{\mathtt{m}(\tau) \mid \exists\Gamma \models \rho.\Gamma \vdash V\!:\!\tau\}$.

$$\begin{aligned} [\![[T,V]]\!]_\rho &= \text{ if } [\![V]\!]_\rho \neq \bot \text{ then } [\![T]\!]_\rho \text{ else } \bot \\ &= \text{ if } \exists\Gamma \models \rho.\Gamma \vdash V\!:\!\sigma \text{ for some } \sigma \text{ then } \bigsqcup\{\mathtt{m}(\tau) \mid \exists\Gamma \models \rho.\Gamma \vdash T\!:\!\tau\} \\ &\qquad \text{else } \bot \text{ by the induction hypothesis} \\ &= \bigsqcup\{\mathtt{m}(\tau) \mid \exists\Gamma \models \rho.\Gamma \vdash [T,V]\!:\!\tau\} \text{ by Lemma 3.1(iv).} \end{aligned}$$

□

Let $\rho_\top$ be the environment which associates $\top$ to all variables. We can characterise strongly normalising and persistently strongly normalising λ^*-terms in the model $\mathcal{HL}_\infty$ as the λ^*-terms whose meaning in the environment $\rho_\top$ is different from $\bot$ and equal to $\top$, respectively. I.e. we have:

Theorem 3.3 (Main Theorem) (i) *A* λ^**-term* S *is strongly normalising iff* $[\![S]\!]_{\rho_\top} \neq \bot$.

(ii) *A* λ^**-term* S *is persistently strongly normalising iff* $[\![S]\!]_{\rho_\top} = \top$.

The proof of this theorem uses the above discussed isomorphism between $\mathcal{HL}_\infty$ and $\mathcal{F}$. The theorem in fact can be reformulated as follows:

Theorem 3.4 *Let* $\Gamma_\omega = \{x\!:\!\omega \mid x \in \mathit{Var}\}$.

(i) *A* λ^**-term* $S \in \mathsf{SN}^*$ *iff* $\Gamma_\omega \vdash S : \varphi$.

(ii) *A* λ^**-term* $S \in \mathsf{PSN}^*$ *iff* $\Gamma_\omega \vdash S : \omega$.

The remaining of the present section is devoted to the proof of this theorem. Subsection 3.1 shows the *if parts* of the claims (i) and (ii) by means of a realizability interpretation of intersection types. The *only if parts* of these claims can be shown using the inductive definitions of SN^*_n and PSN^* given in Section 2: this proof is the content of Subsection 3.2.

3.1 Proof of Theorem 3.4 ($\Leftarrow$)

In order to develop the reducibility method we consider Λ^* as the *applicative structure* whose domain is the set of λ^*-terms and whose application is just the application of λ^*-terms.

We first define a mapping between types and sets of λ^*-terms.

Definition 3.5 The *interpretation of types* is the mappping $[\![\ ]\!]$ defined by:

$$\begin{aligned}
[\![\varphi]\!] &= \mathsf{SN}^* \\
[\![\omega]\!] &= \mathsf{PSN}^* \\
[\![\sigma \to \tau]\!] &= \{S \in \Lambda^* \mid \forall T \in [\![\sigma]\!] \;\; ST \in [\![\tau]\!]\} \\
[\![\sigma \cap \tau]\!] &= [\![\sigma]\!] \cap [\![\tau]\!].
\end{aligned}$$

We extend to Λ^* the standard definition of *saturated set*, as given for example in Krivine [6], [7].

Definition 3.6 A set $\mathcal{S} \subseteq \Lambda^*$ is *saturated* if for all $S, T, \vec{U} \in \Lambda^*$:

$$\begin{aligned}
&S[x := T]\vec{U} \in \mathcal{S} \;\&\; x \in FV(S) \Rightarrow (\lambda x.S)T\vec{U} \in \mathcal{S} \\
&[S, T]\vec{U} \in \mathcal{S} \;\&\; x \notin FV(S) \Rightarrow (\lambda x.S)T\vec{U} \in \mathcal{S} \\
&S\vec{U} \in \mathcal{S} \;\&\; T \in \mathsf{SN}^* \Rightarrow [S, T]\vec{U} \in \mathcal{S}.
\end{aligned}$$

We can show that all sets in the range of our interpretation of types are saturated.

Lemma 3.7 *For all types* τ *the set* $[\![\tau]\!]$ *is saturated.*

Proof. The proof is by structural induction on types. The third condition for atomic types follows from Lemmas 2.2 and 2.3(i).

The more interesting case is that of arrow types. Suppose $S[x := T]\vec{U} \in [\![\tau \to \sigma]\!]$ and $x \in FV(S)$. Let $V \in [\![\tau]\!]$ be arbitrary. By Definition 3.5 $S[x := T]\vec{U}V \in [\![\sigma]\!]$. Then by the induction hypothesis $(\lambda x.S)T\vec{U}V \in [\![\sigma]\!]$. Since V was arbitrary, according to Definition 3.5 we get $(\lambda x.S)T\vec{U} \in [\![\tau \to \sigma]\!]$. Similarly one can show the remaining two conditions. □

The preorder on types agrees with the set theoretic inclusion between type interpretations.

Lemma 3.8 *If $\tau \leq \sigma$, then $[\![\tau]\!] \subseteq [\![\sigma]\!]$.*

Proof. By induction on the length of the derivation of $\tau \leq \sigma$. The definition of PSN^* and Lemma 2.3(ii) justify the axioms for atomic types. □

We define the *valuation of λ^*-terms* $[\![-]\!]_\theta : \Lambda^* \to \Lambda^*$ and the *semantic satisfiability relation* $\models$, which connects the type interpretation and the term valuation, as follows.

Definition 3.9 *Let $\theta : Var \to \Lambda^*$ be a valuation of term variables in Λ^*. Then*

(i) $[\![-]\!]_\theta : \Lambda^* \to \Lambda^*$ *is defined by*
$[\![S]\!]_\theta = S[x_1 := \theta(x_1), \ldots, x_n := \theta(x_n)]$*, where* $FV(S) = \{x_1, \ldots, x_n\}$*;*

(ii) $\theta \models S : \tau$ *if* $[\![S]\!]_\theta \in [\![\tau]\!]$*;*

(iii) $\theta \models \Gamma$ *if* $(\forall (x : \tau) \in \Gamma)$ $\theta \models x : \tau$*;*

(iv) $\Gamma \models S : \tau$ *if* $(\forall \theta \models \Gamma)$ $\theta \models S : \tau$*.*

We can prove that our type assignment system is *sound* for the above semantic satisfiability.

Theorem 3.10 (Soundness)

$$\Gamma \vdash S : \tau \Rightarrow \Gamma \models S : \tau.$$

Proof. By induction on the derivation of $\Gamma \vdash S : \tau$.

Case 1. The last step is (Ax), i.e. $\Gamma, x : \tau \vdash x : \tau$. Then $\Gamma, x : \tau \models x : \tau$ by Definition 3.9(iii).

Case 2. The last step is ([,]), i.e. $\Gamma \vdash S : \tau, \Gamma \vdash T : \sigma \Rightarrow \Gamma \vdash [S, T] : \tau$. By the induction hypothesis $[\![S]\!]_\theta \in [\![\tau]\!]$ and $[\![T]\!]_\theta \in [\![\sigma]\!]$, which implies $[\![T]\!]_\theta \in \mathsf{SN}^*$ by Lemma 3.8. We conclude by Lemma 3.7.

Case 3. The last step is ($\to$ E), i.e. $\Gamma \vdash S : \tau \to \sigma,\ \Gamma \vdash T : \tau \Rightarrow \Gamma \vdash ST : \sigma$. Then by the induction hypothesis $\Gamma \models S : \tau \to \sigma$ and $\Gamma \models T : \tau$. Let $\theta \models \Gamma$, then $[\![S]\!]_\theta \in [\![\tau \to \sigma]\!]$ and $[\![T]\!]_\theta \in [\![\tau]\!]$. Therefore $[\![ST]\!]_\theta \equiv [\![S]\!]_\theta [\![T]\!]_\theta \in [\![\sigma]\!]$.

Case 4. The last step is ($\to$ I), i.e. $\Gamma, x : \tau \vdash S : \sigma \Rightarrow \Gamma \vdash \lambda x.S : \tau \to \sigma$. By the induction hypothesis $\Gamma, x : \tau \models S : \sigma$. Let $\theta \models \Gamma$ and let $T \in [\![\tau]\!]$. We can assume $x \notin \theta(y)$ for all $y \in FV(S)$.
If $x \in FV(S)$, we define $\theta[x := T](x) = T, \theta[x := T](y) = \theta(y)$ for $x \neq y$. Then $\theta[x := T] \models \Gamma$, since $x \notin \Gamma$, and $\theta[x := T] \models x : \tau$, since $T \in [\![\tau]\!]$. Therefore $\theta[x := T] \models S : \sigma$, i.e. $[\![S]\!]_{\theta[x := T]} \in [\![\sigma]\!]$, which means by Definition 3.9(i) that $S[\vec{y} := \theta(\vec{y})][x := T] \in [\![\sigma]\!]$, where $\vec{y} = FV(S) \setminus \{x\}$. By Lemma 3.7 we have $(\lambda x.S[\vec{y} := \theta(\vec{y})])T \in [\![\sigma]\!]$. Then $[\![\lambda x.S]\!]_\theta T \in [\![\sigma]\!]$, since $x \notin FV(\lambda x.S)$. We conclude $[\![\lambda x.S]\!]_\theta \in [\![\tau \to \sigma]\!]$, since $T \in [\![\tau]\!]$ was arbitrary.
If $x \notin FV(S)$, notice that by Lemma 3.8 $T \in [\![\tau]\!]$ implies $T \in \mathsf{SN}^*$. Therefore from $[\![S]\!]_\theta \in [\![\sigma]\!]$ by Lemma 3.7 we get $[[\![S]\!]_\theta, T] \in [\![\sigma]\!]$, which implies $(\lambda x.[\![S]\!]_\theta)T \in [\![\sigma]\!]$ by the same lemma. We conclude $[\![\lambda x.S]\!]_\theta \in [\![\tau \to \sigma]\!]$, since $T \in [\![\tau]\!]$ was arbitrary.

Case 5. The last step is ($\cap$I), i.e. $\Gamma \vdash S : \tau,\ \Gamma \vdash S : \sigma \Rightarrow \Gamma \vdash S : \tau \cap \sigma$. Then by the induction hypothesis $\Gamma \models S : \tau$ and $\Gamma \models S : \sigma$. Let $\theta \models \Gamma$, then $[\![S]\!]_\theta \in [\![\tau]\!]$ and

$[\![S]\!]_\theta \in [\![\sigma]\!]$. Therefore $[\![S]\!]_\theta \in [\![\tau \cap \sigma]\!]$, i.e. $\Gamma \models S : \tau \cap \sigma$.

Case 6. The last step is $(\leq)$, i.e. $\Gamma \vdash S : \tau,\ \tau \leq \sigma \Rightarrow \Gamma \vdash S : \sigma$. By the induction hypothesis $\Gamma \models S : \tau$. Let $\theta \models \Gamma$, then $[\![S]\!]_\theta \in [\![\tau]\!]$. According to Lemma 3.8 $[\![\tau]\!] \subseteq [\![\sigma]\!]$, so it follows that $[\![S]\!]_\theta \in [\![\sigma]\!]$, i.e. $\Gamma \models S : \sigma$. □

Proof of Theorem 3.4($\Leftarrow$) . Let $\Gamma_\omega \vdash S : \varphi$. By soundness (Theorem 3.10) we have that if $\theta \models \Gamma_\omega$, then $[\![S]\!]_\theta \in [\![\varphi]\!] = \mathsf{SN}^*$. We can take $\theta_1(x) = x$, being $\theta_1 \models \Gamma_\omega$, because all variables belong to PSN^*. Obviously, $\theta_1(S) = S$ for every λ^*-term S. Therefore we get that $S \in \mathsf{SN}^*$. Similarly from $\Gamma_\omega \vdash S : \omega$ we get $S \in \mathsf{PSN}^*$. □

Notice that this proof is an extension of the proof given in [4] for Λ to Λ^*.

3.2 *Proof of Theorem 3.4 ($\Rightarrow$)*

It is useful to have the invariance of typing under subject expansion. This property has been proved in [4] for Λ.

Theorem 3.11 (Subject Expansion) (i) *If $\Gamma \vdash S[x := T] : \tau$ and $x \in FV(S)$, then $\Gamma \vdash (\lambda x.S)T : \tau$.*

(ii) *If $\Gamma \vdash [S, T] : \tau$ and $x \notin FV(S)$, then $\Gamma \vdash (\lambda x.S)T : \tau$.*

Proof. The proof of (i) done in [4] for Λ extends to Λ^*.
For (ii) let $\Gamma \vdash [S, T] : \tau$. By Lemma 3.1(iv) we get $\Gamma \vdash S : \tau$ and $\Gamma \vdash T : \sigma$ for some type σ. Since $x \notin FV(S)$ by strengthening and weakening we derive $\Gamma, x : \sigma \vdash S : \tau$. We conclude using rules $(\to \mathrm{I})$ and $(\to \mathrm{E})$. □

The proof of Theorem 3.4 ($\Rightarrow$) can be done using the inductive definitions of SN^*_n and PSN^* given in Section 2. More precisely it easily follows from the following lemma, whose restriction to Λ is proved in [9].

Lemma 3.12 (i) *If $S \in \mathsf{PSN}^\sharp$, then $\Gamma_\omega \vdash S : \omega$.*

(ii) *If $S \in \mathsf{SN}^\sharp_n$, then $\Gamma_\omega \vdash S : \varphi^n \to \varphi$ where $\varphi^n \to \varphi = \underbrace{\varphi \to \ldots \varphi \to}_{n} \varphi$.*

Proof. The proof of the same Lemma done in [9] for Λ considers all rules in Figure 1 but those in the lines four, six, seven, and eight, so we only need to consider these rules.

For the rule

$$\frac{\lambda\vec{x}.S[y := T]\vec{V} \in \mathsf{SN}^\sharp_n \quad y \in FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{SN}^\sharp_n}$$

by the induction hypothesis we know that $\Gamma_\omega \vdash \lambda\vec{x}.S[y := T]\vec{V} : \varphi^n \to \varphi$. Let $m = \mathrm{lh}(\vec{x})$. We suppose $m \leq n$, since the case $m > n$ is similarly proved. By Lemma 3.1(iii) we get $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash S[y := T]\vec{V} : \varphi^{n-m} \to \varphi$. By Lemma 3.1(ii) there exists $\vec{\tau}$ such that $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash S[y := T] : \vec{\tau} \to \varphi^{n-m} \to \varphi$ and $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash V_i : \tau_i$ for $1 \leq i \leq h$, where $h = \mathrm{lh}(\vec{V})$. This implies $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash (\lambda y.S)T : \vec{\tau} \to \varphi^{n-m} \to \varphi$ by Theorem 3.11(i) and so we conclude $\Gamma_\omega \vdash \lambda\vec{x}.(\lambda y.S)T\vec{V} : \varphi^n \to \varphi$ by the rules $(\to \mathrm{E})$ and $(\to \mathrm{I})$.

For the rule

$$\frac{\lambda\vec{x}.[S,T]\vec{V} \in \mathsf{SN}^\sharp_n \quad y \notin FV(S)}{\lambda\vec{x}.(\lambda y.S)T\vec{V} \in \mathsf{SN}^\sharp_n}$$

by the induction hypothesis we know that $\Gamma_\omega \vdash \lambda\vec{x}.[S,T]\vec{V} : \varphi^n \to \varphi$. Let $m = \mathrm{lh}(\vec{x})$. We suppose $m \leq n$, since the case $m > n$ is similarly proved. By Lemma 3.1(iii) we get $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash [S,T]\vec{V} : \varphi^{n-m} \to \varphi$. By Lemma 3.1(ii) there exists $\vec{\tau}$ such that $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash [S,T] : \vec{\tau} \to \varphi^{n-m} \to \varphi$ and $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash V_i : \tau_i$ for $1 \leq i \leq h$, where $h = \mathrm{lh}(\vec{V})$. This implies $\Gamma_\omega, \vec{x} : \vec{\varphi} \vdash (\lambda y.S)T : \vec{\tau} \to \varphi^{n-m} \to \varphi$ by Theorem 3.11(ii) and so we conclude $\Gamma_\omega \vdash \lambda\vec{x}.(\lambda y.S)T\vec{V} : \varphi^n \to \varphi$ by the rules ($\to$ E) and ($\to$ I).

For the rule

$$\frac{\lambda\vec{x}.S\vec{T} \in \mathsf{SN}^\sharp_n \quad \lambda\vec{x}.U \in \mathsf{SN}^\sharp_m \quad \mathrm{lh}(\vec{x}) = m}{\lambda\vec{x}.[S,U]\vec{T} \in \mathsf{SN}^\sharp_n}$$

by the induction hypothesis we know that $\Gamma_\omega \vdash \lambda\vec{x}.S\vec{T} : \varphi^n \to \varphi$ and $\Gamma_\omega \vdash \lambda\vec{x}.U : \varphi^m \to \varphi$. We suppose $m > n$, since the case $m \leq n$ is similarly proved. Let $\vec{x}_1\vec{x}_2 = \vec{x}$ and $\mathrm{lh}(\vec{x}_1) = n$. By Lemma 3.1 we have $\Gamma_\omega, \vec{x}_1 : \vec{\varphi}, \vec{x}_2 : \vec{\omega} \vdash S : \vec{\tau} \to \varphi$, and $\Gamma_\omega, \vec{x}_1 : \vec{\varphi}, \vec{x}_2 : \vec{\omega} \vdash T_i : \tau_i$ $(1 \leq i \leq \mathrm{lh}(\vec{T}))$, and $\Gamma_\omega, \vec{x}_1 : \vec{\varphi}, \vec{x}_2 : \vec{\omega} \vdash U : \varphi$. We conclude using the rules ([,]), ($\to E$), and ($\to I$). The proofs for the remaining rules are similar. □

4 Concluding remarks

We have shown that for a λ^*-term S the following four conditions are equivalent:

(i) $SX_1 \ldots X_n$ is strongly normalising for all n and all strong normalising $X_1, \ldots, X_n$.

(ii) $S \in \mathsf{PSN}^\sharp$ as defined in Figure 1.

(iii) $\Gamma_\omega \vdash S : \omega$ in the intersection type assignement system $\mathcal{HL}^*$.

(iv) $[\![S]\!]_{\rho_\top} = \top$ in the model $\mathcal{HL}_\infty$.

As an application of the "Substitution Theorem" we get that:

$$\exists X_1, \ldots, X_n \in \mathsf{SN}^*.SX_1 \ldots X_n \notin \mathsf{SN}^* \;\Rightarrow \exists X \in \mathsf{SN}^*.S\underbrace{X \ldots X}_{n} \notin \mathsf{SN}^*$$

therefore we plan to investigate consequences of this theorem in the study of infinite reductions.

References

[1] Samson Abramsky. Domain Theory in Logical Form. *Ann. Pure Appl. Logic*, 51(1-2):1–77, 1991.

[2] Gérard Boudol. On Strong Normalization in the Intersection Type Discipline. In Martin Hofmann, editor, *TLCA'03*, volume 2701 of *LNCS*, pages 60 –74. Springer-Verlag, 2003.

[3] Mariangiola Dezani-Ciancaglini, Furio Honsell, and Yoko Motohama. Compositional Characterization of λ-terms using Intersection Types. *Theoret. Comput. Sci.*, 340(3):459–495, 2005.

[4] Furio Honsell and Marina Lenisa. Semantical Analysis of Perpetual Strategies in λ-calculus. *Theoret. Comput. Sci.*, 212(1-2):183–209, 1999.

[5] Jan Willem Klop. *Combinatory Reduction Systems*. PhD thesis, Utrecht University, 1980. Appeared as Mathematical Centre Tracts 127, Kruislaan 413, 1098 SJ Amsterdam.

[6] Jean-Louis Krivine. *Lambda-calcul, types et modèles*. Masson, Paris, 1990.

[7] Jean-Louis Krivine. *Lambda-calculus, types and models*. Ellis Horwood, New York, 1993. Translated from the 1990 French original by René Cori.

[8] Robert Pieter Nederpelt. *Strong Normalisation for a Typed Lambda Calculus with Lambda Structured Types*. PhD thesis, Eindhoven University, 1973.

[9] Makoto Tatsuta and Mariangiola Dezani-Ciancaglini. Normalisation is Insensible to Lambda-term Identity or Difference. In *LICS'06*, 2006. to appear.